

AMENDMENTS TO THE CLAIMS

This listing of claims replaces all prior versions, and listings, of claims in the application:

1 1. (Cancelled)

1 2. (Previously Presented) The method of claim 7, wherein receiving a data
2 transaction request comprises receiving a request for loading data into the database.

1 3. (Previously Presented) The method of claim 7, wherein receiving a data
2 transaction request comprises receiving a request to perform a data transformation operation
3 upon the data in the database.

1 4. (Original) The method of claim 3, wherein receiving a request to perform the data
2 transformation operation comprises receiving a request to perform one of a data selection
3 operation, a data validation operation, a data cleansing operation, and a data query operation.

1 5. – 6. (Cancelled)

1 7. (Currently Amended) A method of performing parallel data operations upon data
2 in a database, comprising:

3 receiving a data transaction request in a client system;
4 executing each of a plurality of multi-phase parallel tasks in plural phases in
5 response to the request to perform the data operations upon the data in the database, ~~wherein~~
6 ~~executing the multi-phase parallel tasks comprises executing each of the parallel tasks in plural~~
7 ~~phases; and~~
8 each parallel task providing a code to indicate if the task is to be re-invoked in the
9 next phase,

10 wherein executing the plurality of multi-phase parallel tasks comprises:
11 executing at least first and second software components in parallel;
12 each of the first and second software components performing one or more
13 operations in a first phase;
14 waiting for a message comprising the code from each of the first and second
15 software components prior to proceeding to a second phase; and
16 each of the first and second software components performing one or more
17 operations in the second phase.

1 8. (Original) The method of claim 7, wherein providing the code comprises
2 providing the code to a task coordinator.

1 9. - 10. (Cancelled)

1 11. (Currently Amended) A method of performing parallel data operations upon data
2 in a database, comprising:

3 receiving a data transaction request in a client system;
4 executing a plurality of multi-phase parallel tasks in response to the request to
5 perform the data operations upon the data in the database;
6 analyzing the transaction request;
7 creating a task plan in response to the transaction request;
8 implementing the task plan in a multi-phase organization, wherein the plurality of
9 multi-phase parallel tasks are executed to implement the task plan;
10 determining, by a task coordinator, whether an additional phase is required to
11 execute the tasks based on codes returned by the tasks to the task coordinator; and
12 scheduling, by the task coordinator, an additional phase in response to the
13 determination that an additional phase is required;
14 re-invoking, by the task coordinator, a first one of the parallel tasks in the
15 additional phase in response to the first parallel task providing a first code indicating the first
16 parallel task is to be re-invoked,
17 wherein the task coordinator does not re-invoke a second one of the parallel tasks
18 in the additional phase in response to the second parallel task providing a second code indicating
19 the second parallel task is not to be re-invoked.

1 12. (Original) The method of claim 11, wherein implementing the task plan comprises
2 creating a job script.

1 13. (Previously Presented) The method of claim 11, wherein implementing the task
2 plan comprises:
3 translating the task plan;
4 selecting a plurality of software components corresponding to the parallel tasks to
5 implement the translated task plan;
6 assigning a plurality of processes corresponding to the software components; and
7 creating a communications channel to allow for communications between the
8 processes.

1 14. (Previously Presented) The method of claim 13, wherein selecting the plurality of
2 software components to implement the translated task plan comprises selecting the plurality of
3 software components to perform at least one of a data extraction operation, a data transformation
4 operation, and a data loading operation.

1 15. (Currently Amended) An apparatus, comprising:
2 a user interface;
3 a processor coupled with the user interface, wherein the processor receives a data
4 transaction request from the user interface; and
5 a controller coupled with the processor, wherein the controller performs a
6 plurality of tasks in parallel based upon instructions received from the processor, each task
7 performed in a plurality of phases,
8 each task to provide a code to indicate whether the task is to be re-invoked in a
9 next phase,
10 wherein the controller comprises at least first and second software components
11 executable in parallel to perform the plurality of tasks;
12 wherein each of the first and second software components is executable to
13 perform one or more operations in a first phase;
14 the controller to wait for a message comprising the code from each of the first and
15 second software components prior to proceeding to a second phase; and
16 wherein each of the first and second software components is executable to
17 perform one or more operations in the second phase.

1 16. (Original) The apparatus of claim 15, wherein the processor generates a task plan
2 in response to the data transaction request.

1 17. (Original) The apparatus of claim 16, wherein the controller comprises a task
2 coordinator to execute the task plan.

1 18. (Original) The apparatus of claim 16, wherein the controller further comprises a
2 plurality of components to implement the task plan in parallel.

1 19.-20. (Cancelled)

1 21. (Previously Presented) The apparatus of claim 15, wherein the controller performs
2 a number of tasks in parallel based upon instructions received from the processor, each task
3 performed in a plurality of phases further comprises the controller performing the tasks in a
4 sequence of multiple process steps.

1 22. (Currently Amended) A system, comprising:
2 a database system;
3 a network; and
4 a client system separate from the database system and coupled to the database
5 system over the network, the client system to establish plural sessions with the database system
6 ~~to implement a plurality of data operations upon the database system in parallel,~~
7 wherein the client system is adapted to execute plural tasks in parallel, each of the
8 plural tasks executable in plural phases, and each task to provide a code to indicate whether the
9 task is to be re-invoked in a next phase, the client system comprising a task coordinator, the task
10 coordinator to:
11 re-invoke a first one of the plurality of tasks in response to the first task
12 providing a first code to the task coordinator; and
13 not re-invoke a second one of the plurality of tasks in response to the
14 second task providing a second code to the task coordinator.

1 23. (Cancelled)

1 24. (Previously Presented) The system of claim 22, wherein the database system is a
2 parallel database system.

1 25. (Previously Presented) The system of claim 22, wherein the client system
2 comprises:
3 a processor to receive a data transaction request;
4 a plurality of operators to perform parallel data operations in response to the data
5 transaction request;
6 an operator interface coupled to the operators, wherein the operator interface
7 allows communications between the operators.

1 26.-27. (Cancelled)

1 28. (Currently Amended) An article comprising at least one storage medium
2 containing instructions that when executed cause a client system to:
3 receive a data transaction request;
4 establish plural sessions with a database system over the network connection in
5 response to the request; and
6 execute a plurality of parallel tasks in the plural sessions to perform data
7 operations upon the data in the database system over a network connection, wherein the client
8 system is separate from the database system, wherein each of the parallel tasks is executed in
9 plural phases, and wherein executing the parallel tasks in plural phases comprises:
10 executing at least first and second software components in parallel;
11 each of the first and second software components performing one or more
12 operations in a first phase;
13 waiting for a message from each of the first and second software
14 components prior to proceeding to a second phase; and
15 each of the first and second software components performing one or more
16 operations in the second phase.

1 29. (Cancelled)

1 30. (Currently Amended) The article of claim ~~[[29]]~~ 28, wherein the instructions
2 when executed cause the client system to execute a first parallel task in a first number of phases
3 and a second parallel task in a second, different number of phases.

1 31. – 43. (Cancelled)

1 44. (Currently Amended) The method of claim ~~[[43]]~~ 7, further comprising:
2 waiting for another message from each of the first and second software
3 components prior to proceeding to a third phase;
4 the first software component performing one or more operations in the third
5 phase; and
6 the second software component being idle in the third phase.

1 45. (Previously Presented) The method of claim 44, further comprising:
2 receiving a first message from the first software component indicating that the
3 first software component is to be re-invoked in the third phase; and
4 receiving a second message from the second software component indicating that
5 the second component is not to be re-invoked in the third phase.

1 46. (Cancelled)

1 47. (Currently Amended) The apparatus of claim ~~[[46]]~~ 15, wherein the controller is
2 adapted to further wait for another message from each of the first and second software
3 components prior to proceeding to a third phase;
4 wherein the first software component is executable to perform one or more
5 operations in the third phase, and the second software component is idle in the third phase.

1 48. (Previously Presented) The apparatus of claim 47, wherein the controller is
2 adapted to further:

3 receive a first message from the first software component indicating that the first
4 software component is to be re-invoked in the third phase; and

5 receive a second message from the second software component indicating that the
6 second component is not to be re-invoked in the third phase.

1 49. – 50. (Cancelled)

1 51. (Currently Amended) The article of claim [[50]] 28, wherein the instructions
2 when executed cause the client system to further:

3 wait for another message from each of the first and second software components
4 prior to proceeding to a third phase;

5 cause the first software component to perform one or more operations in the third
6 phase; and

7 cause the second software component to be idle in the third phase.

1 52. (Previously Presented) The article of claim 51, wherein the instructions when
2 executed cause the client system to further:

3 receive a first message from the first software component indicating that the first
4 software component is to be re-invoked in the third phase; and

5 receive a second message from the second software component indicating that the
6 second component is not to be re-invoked in the third phase.

1 53. (Previously Presented) The method of claim 7, wherein executing the plurality of
2 multi-phase parallel tasks includes executing a plurality of checkpoint tasks in parallel, each
3 checkpoint task having multiple phases, and each checkpoint task to write data to storage to
4 provide an indication of a current execution point.

1 54. (Previously Presented) The method of claim 7, wherein executing the plurality of
2 multi-phase parallel tasks includes executing the plurality of multi-phase tasks in parallel.

1 55. (Previously Presented) The apparatus of claim 15, wherein the plurality of tasks
2 include a plurality of checkpoint tasks that are executed in parallel, each checkpoint task having
3 multiple phases, and each checkpoint task to write data to storage to provide an indication of a
4 current execution point.

1 56. (Cancelled)

1 57. (Currently Amended) ~~The method of claim 56;~~ A method of performing parallel
2 data operations upon data in a database, comprising:
3 receiving a data transaction request in a client system;
4 executing a plurality of multi-phase parallel tasks in response to the request to
5 perform the data operations upon the data in the database, wherein executing the multi-phase
6 parallel tasks comprises executing each of the parallel tasks in plural phases; and
7 each parallel task providing a code to a task coordinator to indicate if the task is to
8 be re-invoked in the next phase;
9 re-invoking, by the task coordinator, a first one of the parallel tasks in the next
10 phase in response to the first parallel task providing a first code indicating the first parallel task is
11 to be re-invoked,
12 wherein the task coordinator does not re-invoke a second one of the parallel tasks
13 in the next phase in response to the second parallel task providing a second code indicating the
14 second parallel task is not to be re-invoked.

1 58. (Previously Presented) The method of claim 57, wherein executing the plurality of
2 multi-phase parallel tasks comprises executing first and second software components, the first
3 parallel task comprising the first software component, and the second parallel task comprising
4 the second software component,
5 wherein re-invoking the first parallel task comprises re-invoking the first software
6 component.

1 59. – 60. (Cancelled)

1 61. (Previously Presented) The apparatus of claim 17, the task coordinator to:
2 re-invoke a first one of the plurality of tasks in response to the first task providing
3 a first code to the task coordinator; and
4 not re-invoke a second one of the plurality of tasks in response to the second task
5 providing a second code to the task coordinator.